chinery, New York, NY, USA. 2020. Article 1, P. 1–6. DOI: https://doi.org/10.1145/3377283.337.
4. Saetchnikov I., Tcherniavskaia E. A., Skakun V. V. Object detection for unmanned aerial vehicle camera via convolutional neural networks // In IEEE Journal on Miniaturization for Air and Space Systems. V. 2. N. 2. P. 98-103. doi: 10.1109/JMASS.2020.3040976.
5. Saetchnikov I., Skakun V. V., Tcherniavskaia E. A. Pattern recognition on aerospace images using deep neural networks // IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Pisa, Italy. 2020. P. 336-340, doi: 10.1109/MetroAeroSpace48742.2020.9160198.
6. Saetchnikov I., Skakun V. V., Tcherniavskaia E. A. Efficient objects tracking from an unmanned aerial vehicle // IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace). 2021 P. 221-225. doi: 10.1109/MetroAeroSpace51421.2021.9511748.

МАШИННОЕ ОБУЧЕНИЕ ПРИ ПЕРЕМЕЩЕНИИ МОБИЛЬНОГО РОБОТА

А. В. Сидоренко, Н. А. Солодухо

Белорусский государственный университет, Минск, Беларусь

Рассмотрены вопрос моделирования при навигации с огибанием препятствий мобильного робота с использованием методов машинного обучения: Q-обучения, алгоритма SARSA, глубокого Q-обучения и двойного глубокого Q-обучения. Разработанное программное обеспечение включает средства Mobile Robotics Simulation Toolbox, Reinforcement Learning Toolbox и пакет визуализации Gazebo для моделирования среды. Результаты вычислительного эксперимента показывают, что для моделируемой среды размером 17 на 17 блоков и препятствия длиной в 12 блоков обучение при использовании алгоритма SARSA происходит с лучшей производительностью, чем для остальных

Ключевые слова*: робот, машинное обучение, Q-обучение, перемещение.*

ВВЕДЕНИЕ

При внедрении мобильных роботов в космическую, военную, производственную сферы деятельности человека одной из актуальных является проблема управления движением мобильного робота в некоторой среде при известном местоположении робота, расположении целевых точек, в которые должен переместиться робот [1]. При этом существенным является обеспечение безопасного движения робота без столкновения со встречающимися на его пути препятствиями.

При решении подобных задач, как правило, используются алгоритмы машинного обучения, включающие алгоритмы обучения с подкреплением, нейросетевые алгоритмы, алгоритмы глубокого обучения [2, 3]. Ис-

пользование указанных алгоритмов основано на принципах моделирования. Критерием оптимизации в каждой из указанных моделей является определение вознаграждения для обучения алгоритма.

В данной работе целью является сравнение алгоритмов обучения для безопасного движения мобильного робота, демонстрация возможностей созданного программного обеспечения и проведения вычислительного эксперимента при навигации.

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Алгоритм Q-обучения представляет собой метод, используемый при машинном обучении в сфере искусственного интеллекта при мультиагентном подходе. На основе полученного от среды вознаграждения при использовании данного алгоритма агент формирует функцию полезности Q, что в последствии дает ему возможность уже случайным образом выбирать стратегию поведения, а также учитывать опыт предыдущего взаимодействия со средой.

Алгоритм Q-обучения рассматривает пару: состояние-действие. Он представляет собой алгоритм, позволяющий вычислить значение какого-либо действия в определенном состоянии. Для любого процесса принятия решений при Q-обучении определяется оптимальная величина Q.

Алгоритм Q-обучения позволяет агенту получить вознаграждение, совершая в конкретном состоянии наиболее оптимальное действие. Опираясь на таблицу вознаграждений, он позволяет выбрать следующее действие в зависимости от того, насколько оно полезно, и дает возможность агенту обновить величину, называемую Q-значением. Q- величины инициализируются случайными значениями, которые обновляются согласно выражению

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_{\alpha} Q] \quad (1)$$

где a_t и s_t - действие и состояние агента в момент времени t, α и γ - скорость обучения и дисконтирующий множитель, параметры которых находятся в диапазоне [0, 1], r - значение вознаграждения[3]. В результате создается новая таблица, называемая Q- таблицей, в которой хранится информация о состоянии и действии агента. Хранение информации в Q-таблице может дать сбой при значительном увеличении числа состояний/действий.

Алгоритм управления SARSA (State- Action- Reward- State-Action) является вариацией алгоритма Q-обучения. Он основан на методе временных различий и представляет собой набор пар: состояние-действие в некоторой среде с определением действия для каждого состояния агента в мультиагентной системе. При этом производится оценка функции зна-

чения Q(s,a) для всех пар состояние-действие (s,a) на каждом временном шаге на основе правила обновления метода временных различий

$$Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)], \quad (2)$$

где a_t и s_t -действие и состояние агента в момент времени t, α- скорость обучения, γ и r дисконтирующий множитель и значение вознаграждения, α и γ параметры, значения которых находятся в диапазоне [0, 1] [3].

Если состояние s является конечным, то выполняется соотношение $Q(s,a)=1$, $a \subset A$, где A это множество всех возможных действий.

Глубокое Q-обучение. При реализации Q-обучение можно комбинировать с приближением функции. Одним из решений такой задачи является применение нейронной сети в качестве аппроксиматора функции [2]. При этом использовании нейронных сетей особенно полезно при обучении с подкреплением, когда пространство состояний или пространство действий слишком велико. Нейронная сеть может использоваться для аппроксимации функции значения или пары: действие-состояние в значении Q.

Как и все нейронные сети, они используют коэффициенты для аппроксимации функции, связывающей входы с выходами, и их обучение заключается в поиске оптимальных коэффициентов или весов путем итеративной корректировки этих весов вдоль градиентов, которые обещают меньшую ошибку [3].

Двойное глубокое Q-обучение работает так же, как и глубокое Q-обучение, с той лишь разницей, что текущее значение действия и текущая политика выбора действий разделены, при этом целесообразно использовать две отдельные функции значения ценности Q. Практически, с использованием разных опытов симметрично другу обучаются две отдельные функции значения ценности Q. В задаче решения выбора действия вычисляется среднее двух функций ценности и на основании полученного значения принимается решение о совершении того или иного действия.

ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Программно реализованные алгоритмы обучения, примененные к разработанной нами модели управления мобильным роботом, позволили провести вычислительный эксперимент с использованием разработанной компьютерной программы. Для реализации алгоритма обучения использовался пакет для Matlab Reinforcement Learning Toolbox [4]. В модели, описывающей движение робота, применялся пакет Mobile Robotics Simulation Toolbox [5] на операционной системе Linux при использовании пакета визуализации среды Gazebo. Взаимодействие агентов обеспе-

чивается через пакет для Matlab ROS Toolbox [6]. При проведении вычислительного эксперимента в качестве среды использовалась поверхность 17 на 17 блоков с препятствием в 12 блоков (рис. 1а). Для достижения защиты от ошибочного пересечения препятствий блоки стены из Gazebo в симуляции Matlab были окружены дополнительным слоем стены сверху, снизу и справа. В процессе эксперимента при перемещении робота в конечную (целевую) точку, вознаграждение, равное «500», определялось как целевое. Перемещение робота в любое другое местоположение определялось значением вознаграждением в «-1». Обучение прекращалось, когда суммарное среднее (среднее значение за последние 30 эпизодов обучения) достигало значения вознаграждения «480».

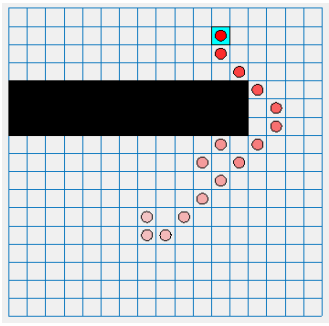
При выполнении исследования мультиагентная система проходила оптимальный путь следования при огибании препятствия и использовании для обучения приведенных выше алгоритмов. Результаты вычислительного эксперимента приведены на рис. 1б.

Анализ полученных алгоритмов показал, что для среды размером 17 на 17 блоков с препятствием длиной в 12 блоков обучение производится при использовании алгоритма SARSA за 58 эпизодов, при Q-обучение - за 73 эпизода, при алгоритме глубокого обучения за 147 эпизодов, а для алгоритма двойного глубокого обучения понадобилось соответственно 236 эпизодов. При этом длина траектории, описываемая перемещающимся роботом при использовании алгоритмов SARSA, Q-обучения и двойного глубокого Q-обучения составила 16 шагов, для глубокого Q-обучения, соответственно, 15 шагов.

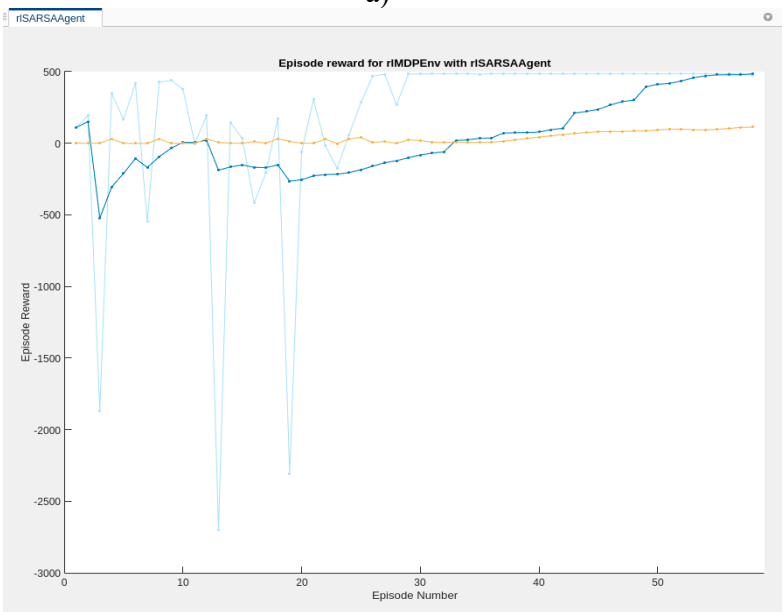
ЗАКЛЮЧЕНИЕ

В процессе выполнения исследований проанализирована работа алгоритмов машинного обучения с подкреплением для безопасного движения мобильного робота: SARSA, Q-обучение, глубокое Q-обучение и двойное глубокое Q-обучение. Разработано программное обеспечение, проведен вычислительный эксперимент по обучению роботизированной системы.

Результаты анализа показали, что быстрее всего обучается робот при использовании алгоритма SARSA, а медленнее всего - алгоритма глубокого двойного Q-обучения. Длина траекторий движения робота для всех алгоритмов была приблизительно одинаковой.



а)



б)

Рис.1. Вид симуляции движения робота в Matlab (а) и зависимости значений вознаграждений от количества итераций (б) при использовании алгоритма SARSA

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Назарова А. В., Рыжова Т.П. Методы и алгоритмы мультиагентного управления робототехнической системой // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2012. Спец. вып. 6 : Робототехнические системы. С. 93-105.
2. Fu Y., et al. Neural Network-Based Learning from Demonstration of an Autonomous Ground Robot // Machines. 2019. V. 7. №2. P. 1-14.
3. Altuntas N., et al. Reinforcement learning based mobile robot navigation // Turkish Journal of electrical engineering & Computer sciences. 2016. V. 24. №3. P. 1747-1767.
4. Описание пакета Reinforcement Learning Toolbox [Электронный ресурс]. – Режим доступа: https://www.mathworks.com/help/reinforcement-learning/index.html?s_tid=CRUX_lftnav . – Дата доступа: 17.03.2021.
5. Описание пакета Mobile Robotics Simulation Toolbox [Электронный ресурс]. – Режим доступа: https://www.mathworks.com/matlabcentral/fileexchange/66586-mobile-robotics-simulation-toolbox. – Дата доступа: 17.03.2021.
6. Описание пакета ROS Toolbox [Электронный ресурс]. – Режим доступа: https://www.mathworks.com/products/ros.html. – Дата доступа: 17.03.2021.